
NLSY97 Appendix 3: Family Background and Formation Variable Creation

HOUSEHOLD SIZE AS OF THE SURVEY DATE

Variables Created:

CV_HH_SIZE
CV_HH_UNDER_6
CV_HH_UNDER_18

Variables Used

Name in Program	Question Name on CD
PUBID	PUBID
HAGE01-HAGE14	HHI_AGE.01-.14
HUID01-HUID14	HHI_UID.01-.14
RSAGE	SYMBOL!KEY!AGE
H5070001-H5070010	YHHI-50700.01-.10

This program creates several variables describing the composition of the respondent's household: the total number of residents, the number of residents under age 6, and the number of residents under age 18.

```

array hage hage01-hage14;
array huid huid01-huid14;

/* Create dummy variables hhdum[i] (i=1 TO 14)
   that equals one if the ith member has a member ID,
   and zero if the ith member does not have a member
   ID (i.e.hhid0i=-4). Also create dummies for
   members under age 6 and under age 18.*/

array hhdum hhdum01-hhdum14;
array dum6 dum601-dum614;
array dum18 dum1801-dum1814;

do i=1 to 14;
  hhdum[i]=0;
  dum6[i]=0;
  dum18[i]=0;
end;

do i=1 to 14;
  if huid[i]>-1 then hhdum[i]=1;
  if -1<hage[i]<6 then dum6[i]=1;
  if -1<hage[i]<18 then dum18[i]=1;
end;

/* Create the dummy for the respondent's age */
rdum6=0;      rdum18=0;

if -1<rsage<6 then rdum6=1;
if -1<rsage<18 then rdum18=1;

/* Create household size hysize by adding up the
   dummies hhdum[i] and also add one for the
   respondent. Similarly, create variabes under6 and
   under18 by adding up the other two dummies.*/
hysize=1;
under6=rdum6;
under18=rdum18;

do i=1 to 14;
  hsize=hsize+hhdum[i];
  under6=under6+dum6[i];
  under18=under18+dum18[i];
end;

if huid01=-5 then do;
  hsize=-5;
  under6=-5;
  under18=-5;
end;

if -4<huid01<0 or -4<huid02<0 or -4<huid03<0 or
-4<huid04<0 or -4<huid05<0 or -4<huid06<0 or
-4<huid07<0 or -4<huid08<0 or -4<huid09<0 or
-4<huid10<0 or -4<huid11<0 or -4<huid12<0 or
-4<huid13<0 or -4<huid14<0 then hsize=-3;

if -4<hage01<0 or -4<hage02<0 or -4<hage03<0 or
-4<hage04<0 or -4<hage05<0 or -4<hage06<0 or
-4<hage07<0 or -4<hage08<0 or -4<hage09<0 or
-4<hage10<0 or -4<hage11<0 or -4<hage12<0 or
-4<hage13<0 or -4<hage14<0 or -4<rsage<0 then do;
  under6=-3;
  under18=-3;
end;

/*There are four households with one member with
an estimated age under 18. Add 1 in each case.*/
if h5070001 in (1,2) or h5070002 in (1,2) or
h5070003 in (1,2) or h5070004 in (1,2) or
h5070005 in (1,2) or h5070006 in (1,2) or
h5070007 in (1,2) or h5070008 in (1,2) or
h5070009 in (1,2) or h5070010 in (1,2) then
under18=under18+1;

endsas;

```

YOUTH'S RELATIONSHIP TO HOUSEHOLD PARENT FIGURE(S)

Variables Created: CV_YTH_REL_HH_CURRENT

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
PUBID	PUBID	HH2UID01-HH2UID14	HHI_UID.01-14
YID	YOUTH_ID.01	MARRY_1-MARRY_14	HHI_MARSTAT.01-14
SH931-SH935	SH-93.01-.05 (round 1)	REL_1-REL_14	HHI_RELY.01-14
HHI2_101-HHI2_116	HHI2_HHI1D.01-.16 (round 1)	GEND_1-GEND_14	HHI_SEX.01-14
HH1UID01-HH1UID16	HHI2_UID.01-.16 (round 1)	AGE	CV AGE INT DATE
HHAGE01-HHAGE14	HHI AGE.01-14		

Codes for Created Variable

- | | |
|------------------------------------|---------------------------------|
| 1 = Both biological parents | 6 = Adoptive parent(s) |
| 2 = Two parents, biological mother | 7 = Foster parent(s) |
| 3 = Two parents, biological father | 8 = No parents, grandparents |
| 4 = Biological mother only | 9 = No parents, other relatives |
| 5 = Biological father only | 10 = Anything else |

This program creates a variable identifying the youth's relationship to the primary adults in the household. In round 2 there is no information collected on the legal guardian of the youth, so it is not possible to determine whether respondents are living with non-parent relatives because they are guardians or because the living situation is better (e.g., closer to school, no rent). For this reason, youths above the age 18 and above are considered independent and, if they are not living with an identified parent or parent-figure (legal guardian), are put into the anything else category. Youths below the age of 18 who are not living with an identified parent or parent-figure are put into the category that most closely matches their household situation.

```

array rel_a (i) rel_1-rel_14;
array marry_a (i) marry_1-marry_14;
array gend_a (i) gend_1-gend_14;
array age_a (i) hhage01-hhage14;

drop hh2_101-hh2_116;

*this part determines legal guardians so that we can determine whether any guardians are present from the R1 interview;
array hh1uid (i) hh1uid01-hh1uid14;           *round 1 variable;
array hh2uid (i) hh2uid01-hh2uid14;           *round 2 variable;

do i=1 to 14;
  if yid=1 and sh931>0 and sh931+100=hh1uid then do;
    uid=hh1uid; end;
  if yid=2 and sh932>0 and sh932+100=hh1uid then do;
    uid=hh1uid; end;
  if yid=3 and sh933>0 and sh933+100=hh1uid then do;
    uid=hh1uid; end;
  if yid=4 and sh934>0 and sh934+100=hh1uid then do;
    uid=hh1uid; end;
  if yid=5 and sh935>0 and sh935+100=hh1uid then do;
    uid=hh1uid; end;
  end;

do i=1 to 14;
  if (uid ne . and uid=hh2uid) then line=i;
  if (uid ne . and uid=hh2uid) then line=i;

```

```

if (uid ne . and uid=hh2uid) then line=i;
if (uid ne . and uid=hh2uid) then line=i;
if (uid ne . and uid=hh2uid) then line=i;
end;
legal=0;
do i=1 to 14;
  if line=i then legal=rel_a;
end;

momid=0;      domid=0;      adopdad=0;
admom=0;      fostma=0;      fostda=0;
stepma=0;      stepda=0;      husb=0;
wife=0;        grand=0;       relat=0;
nonrel=0;      indep=0;      spouse=0;
**legal;
do i=1 to 14;
  if (legal>28 and legal<37) then do; grand=1; end;
  *spouse;
  if legal=1 or legal=2 then do; spouse=1; end;
  *brother/sister;
  if (legal>12 and legal<19) and age_a>20 then do; relat=1; end;
  *aunt or uncle and other relatives;
  if (legal>69 and legal<85) and age_a>20 then do; relat=1; end;
  *lover, roommate, other non-relative, mom's or dad's partner;
  if legal=69 or legal=68 or legal=85 or legal=88 or legal=89 then do; nonrel=1; end;
  if legal=-1 or legal=-2 or legal=-3 then do; invalid=1; end;
  if rel_1=-4 and rel_2=-4 and rel_3=-4 and rel_4=-4 and rel_5=-4 and rel_6=-4 and rel_7=-4 and
    rel_8=-4 and rel_9=-4 and rel_10=-4 and rel_11=-4 and rel_12=-4 and rel_13=-4 and rel_14=-4 then do;
    indep=1;
  end;
end;
**not legal;
do i=1 to 14;
  if (rel_a>28 and rel_a<37) then do; nlgrand=1; end;
  *spouse;
  if rel_a=1 or rel_a=2 then do; nlspouse=1; end;
  *brother/sister;
  if (rel_a>12 and rel_a<19) and age_a>20 then do; nlrelat=1; end;
  *aunt or uncle and other relatives;
  if (rel_a>69 and rel_a<85) and age_a>20 then do; relat=1; end;
  *lover, roommate, other non-relative, mom's or dad's partner;
  if rel_a=69 or rel_a=68 or rel_a=85 or rel_a=88 or rel_a=89 then do; nlnnrl=1; end;
  if rel_a=-1 or rel_a=-2 or rel_a=-3 then do; nlinv=1; end;
end;

if age<18 then do;
  if nlgrand ne . and legal=0 then grand=nlgrand;
  if nlspouse ne . and legal=0 then spouse=nlspouse;
  if nlrelat ne . and legal=0 then relat=nlrelat;
  if nlnnrl ne . and legal=0 then nonrel=nlnnrl;
  if nlinv ne . and legal=0 then invalid=nlinv;
end;

*for all youths;
if nlspouse ne . then spouse=nlspouse;

do i=1 to 14;

```

Appendix 3: Family Background and Formation Variable Creation

```
if rel_1=3 then momid=1;
/*and so on through rel_14=3 and momid=14*/
end;

do i=1 to 14;
  if rel_1=4 then domid=1;
/*and so on through rel_14=4 and domid=14*/
end;

do i=1 to 14;
  if rel_1=5 then admom=1;
/*and so on through rel_14=5 and admom=14*/
end;

do i=1 to 14;
  if rel_1=6 then adopdad=1;
/*and so on through rel_14=6 and adopdad=14*/
end;

do i=1 to 14;
  if rel_1=7 then stepma=1;
/*and so on through rel_14=7 and stepma=14*/
end;

do i=1 to 14;
  if rel_1=8 then stepda=1;
/*and so on through rel_14=8 and stepda=14*/
end;

do i=1 to 14;
  if rel_1=9 then fostma=1;
/*and so on through rel_14=9 and fostma=14*/
end;

do i=1 to 14;
  if rel_1=10 then fostda=1;
/*and so on through rel_14=10 and fostda=14*/
end;

/*used to determine whether single parent households contain another parent. If so hand edits are done below*/

do i=1 to 14;
  if momid>0 then do;
    if momid=1 and marry_1=1 then husb=1;
/*and so on through momid=14, marry_14=1, and husb=14*/
    end;
  end;

do i=1 to 14;
  if domid>0 then do;
    if domid=1 and marry_1=1 then wife=1;
/*and so on through domid=14, marry_14=1, and wife=14*/
    end;
  end;

rel=-16;
if age>17 then do; rel=10; end;
```

```
if marry_1=-5 then do; rel=-5; end;
if indep=1 then do; rel=10; end;
if invalid=1 then do; rel=-3; end;
if nonrel>0 then do; rel=10; end;
if relat>0 then do; rel=9; end;
if grand>0 and momid=0 and domid=0 then do; rel=8; end;
if spouse=1 then do; rel=10; end;
if fostda>0 or fostma>0 then do; rel=7; end;
if admom>0 or adopdad>0 then do; rel=6; end;
if stepda>0 or stepma>0 then do; rel=10; end;
if domid>0 and momid=0 then do; rel=5; end;
if momid>0 and domid=0 then do; rel=4; end;
if domid>0 and momid=0 then do; if admom>0 or stepma>0 then rel=3; end;
if momid>0 and domid=0 then do; if adopdad>0 or stepda>0 then rel=2; end;
if momid>0 and domid>0 then do; both=1; rel=1; end;

/*hand edits from husb and wife variables*/    if pubid=504 then rel=2;
                                                if pubid=2261 then rel=2;
                                                if pubid=1099 then rel=10;
/*additional hand edits*/                      if pubid=5636 then rel=9;
*round 1 legal guardians not captured by the above;    if pubid=7236 then rel=9;

if rel=-16 then rel=10;

endsas;
```

YOUTH'S MARITAL STATUS AND MARITAL/COHABITATION HISTORY

Variables Created:	CV_MARSTAT CV_FIRST_COHAB_DATE_M CV_FIRST_MARRY_DATE_M CV_FIRST_COHAB_MONTH CV_COHAB_TTL	CV_MARSTAT_COLLAPSED CV_FIRST_COHAB_DATE_Y CV_FIRST_MARRY_DATE_Y CV_FIRST_COHAB_MONTH CV_MARRIAGES_TTL
---------------------------	--	--

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
Round 1 variables:			
M45001_1, M45002_1	YMAR-4500.01, .02	BDATE_D, BDATE_M, BDATE_Y	KEY!BDATE_D, _M, _Y
M54011_1, M54012_1	YMAR-5400.01.01, .02	FCOH_M, FCOH_Y	CV_FIRST_COHAB_DATE_M, _Y
PUBID	PUBID	FCOHM	CV_FIRST_COHAB_MONTH
MARSTAT1	CV_MARSTAT	FMAR_M, FMAR_Y	CV_FIRST_MARRY_DATE_M, _Y
CMARSTA1	CV_MARSTAT_COLLAPSED	FMARM	CV_FIRST_MARRY_MONTH
COH_TTL	CV_COHAB_TTL	LINT_D, LINT_M, LINT_Y	CV_INTERVIEW_DATE_D, _M, _Y
MAR_TTL	CV_MARRIAGES_TTL		
Round 2 variables:			
M620	YMAR-620	M48002M, M48002Y	YMAR-4800.02-M, ~Y
M650	YMAR-650	M48003M, M48003Y	YMAR-4800.03-M, ~Y
M700	YMAR-700	M540011, M540012	YMAR-5400.01.01, .02
M710	YMAR-710	M560012M, M560012Y	YMAR-5600.01.02-M, ~Y
M712	YMAR-712	M570011M, M570011Y	YMAR-5700.01.01-M, ~Y
M714	YMAR-714	M570012M, M570012Y	YMAR-5700.01.02-M, ~Y
M716	YMAR-716	M63001-M63003	YMAR-6300.01-.03
M718	YMAR-718	M700011M, M700011Y	YMAR-7000.01.01-M, ~Y
M729D	YMAR-729D	M727011, M727021	YMAR-7270.01.01, .02.01
M730	YMAR-730	M730011, M730021	YMAR-7300.01.01, .02.01
M740	YMAR-740	M7900111-M7900113	YMAR-7900.01.01.01-.03
M760	YMAR-760	M7900211	YMAR-7900.02.01.01
M1000	YMAR-1000	M810111M, M810111Y	YMAR-8100.01.01.01-M, ~Y
M1500	YMAR-1500	M810112M, M810112Y	YMAR-8100.01.01.02-M, ~Y
M30501-M30503	YMAR-3050.01-.03	M820111M, M820111Y	YMAR-8200.01.01.01-M, ~Y
M31001M, M31001Y	YMAR-3100.01-M, ~Y	M820211M, M820211Y	YMAR-8200.02.01.01-M
M31002M, M31002Y	YMAR-3100.02-M, ~Y	M910011	YMAR-9100.01.01
M31003M, M31003Y	YMAR-3100.03-M, ~Y	M920011M, M920011Y	YMAR-9200.01.01-M, ~Y
M45001-M45003	YMAR-4500.01-.03	M9800111	YMAR-9800.01.01.01
M46501-M46503	YMAR-4650.01-.03	M101111M, M101111Y	YMAR-10100.01.01.01-M, ~Y
M46701, M46702	YMAR-4670.01, .02	M114001-M114003	YMAR-11400.01-.03
M47001-M47003	YMAR-4700.01-.03	INT_D, INT_M, INT_Y	CV_INTERVIEW_DATE~D, ~M, ~Y
M48001M, M48001Y	YMAR-4800.01-M, ~Y		

Codes for Created Variable

Marital/Cohabitation Status

- 1 = never married, cohabiting
- 2 = never married, not cohabiting
- 3 = married, spouse present
- 4 = married, spouse absent
- 5 = separated, cohabiting
- 6 = separated, not cohabiting
- 7 = divorced, cohabiting
- 8 = divorced, not cohabiting
- 9 = widowed, cohabiting
- 10 = widowed, not cohabiting

Collapsed Marital Status

- 0 = never married
- 1 = married
- 2 = separated
- 3 = divorced
- 4 = widowed

This program creates two variables that describe marital status/cohabitation status as of the interview date for respondents age 16 and older. Other respondents are valid skips (-4). Note that later partners take precedence over earlier partners.

The program also creates variables which provide the dates of the youth's first marriage and/or cohabitation in both a continuous month scheme and as actual dates (for more information on the continuous month scheme, see appendix 7 in this document). Summary variables count the total number of marriages and cohabitations for each youth. Note that these variables are available only for youths age 16 and older as of 12/31/96. If a respondent is cohabiting and then marries it is considered both a cohabitation and a marriage. If someone refuses or doesn't know the full date of their marriage or cohabitation, then the spell is counted in the total variables and the date variables are coded -1 or -2 as applicable.

```

if pubid>0;
if m650 ge 0 then m620=m650;

iym=int_y*100+int_m;
dliym=lint_y*100+lint_m;
dlicm=(lint_y-1980)*12+lint_m;
doicm=(int_y-1980)*12+int_m;

/*set up arrays*/
array m (l) m001-m232;
array coh (l) coha001-coha232;
array mars (l) mars001-mars232;

/*initialize values to 0 for all who go through section*/
if m700>-4 then do;
marstat=2;
ttlm=0;
ttlc=0;

do l=1 to 232;
  if dlicm le L le doicm then do;
    m=0; coh=0; end;
  end;

array ysca (t) M31001Y M700011Y;
array yscb (t) M31002Y yscb2;
array yscc (t) M31003Y yscc2;
array msca (t) M31001M M700011M;
array mscb (t) M31002M mscb2;
array mscc (t) M31003M mscc2;
array yeca (t) M48001Y M920011Y;
array yecb (t) M48002Y yecb2;
array yecc (t) M48003Y yecc2;
array meca (t) M48001M M920011M;
array mech (t) M48002M mech2;
array mecc (t) M48003M mecc2;
array ymsca (t) ymsca1 ymsca2;
array ymscb (t) ymscb1 ymscb2;
array ymscc (t) ymscc1 ymscc2;
array ymeca (t) ymeca1 ymeca2;
array ymecb (t) ymecb1 ymecb2;
array ymecc (t) ymecc1 ymecc2;
array csmca (t) csmca1 csmca2;
array csmcb (t) csmcb1 csmcb2;
array csmcc (t) csmcc1 csmcc2;
array cemca (t) cemca1 cemca2;
array cemcb (t) cemcb1 cemcb2;
array cemcc (t) cemcc1 cemcc2;
array cdliia (t) M46701 cdliia2;
array cdlib (t) M46702 cdlib2;
array cdlic (t) cdlic1 cdlic2;
array cbega (t) M47001 cbega2;
array cbegb (t) M47002 cbegb2;
array cbegc (t) M47003 cbegc2;
array cbeg2a (t) M910011 cbeg2a2;
array cbeg2b (t) cbeg2b1 cbeg2b2;
array cbeg2c (t) cbeg2c1 cbeg2c2;
array mbega (t) M45001 mbega2;
array mbegb (t) M45002 mbegb2;
array mbegc (t) M45003 mbegc2;
array ymsma (t) ymsma1-ymsma2;
array ymsmb (t) ymsmb1-ymsmb2;
array ymsmc (t) ymsmc1-ymsmc2;
array csmma (t) csmma1-csmma2;
array csmbb (t) csmbb1-csmbb2;
array csmmc (t) csmmc1-csmmc2;
array csmsa (t) csmsa1-csmsa2;
array csmsb (t) csmsb1-csmsb2;
array csmsc (t) csmsc1-csmsc2;
array ymssa (t) ymssa1-ymssa2;
array ymssb (t) ymssb1-ymssb2;
array ymssc (t) ymssc1-ymssc2;
array mdlia (t) M45001_1 mdlia2;
array mdlib (t) M45002_1 mdlib2;
array mdlic (t) mdlic1 mdlic2;
array cemma (t) cemma1-cemma2;
array cemmb (t) cemmb1-cemmb2;
array cemmc (t) cemmc1-cemmc2;
array howa1 (t) M540011 M9800111;
array howa2 (t) M540012 howa22;
array hyma1 (t) M570011Y M101111Y;
array hmma1 (t) M570011M M101111M;
array hyma2 (t) M570012Y hyma22;
array hmma2 (t) M570012M hmma22;
array hysa2 (t) M560012Y hysa22;
array hmsa2 (t) M560012M hmsa22;
array howa3 (t) M7900111 howa32;
array howb3 (t) M7900211 howb32;
array hyma3 (t) M820111Y hyma32;
array hymb3 (t) M820211Y hymb32;
```

```

array hmma3 (t) M820111M hmma32;
array hmmb3 (t) M820211M hmmb32;
array hysa3 (t) M810111Y hysa32;
array hmsa3 (t) M810111M hmsa32;
array howa4 (t) M7900112 howa42;
array hysa4 (t) M810112Y hysa42;
array hmsa4 (t) M810112M hmsa42;
array howa5 (t) M7900113 howa52;
array cmarsa (t) cmarsa1 cmarsa2;
array cmarsb (t) cmarsb1 cmarsb2;
array cmarsc (t) cmarsc1 cmarsc2;
array nuca (t) nuca1 nuca2;
array nucb (t) nucb1 nucb2;
array nucc (t) nucc1 nucc2;
array numa (t) numa1 numa2;
array numb (t) numb1 numb2;
array numc (t) numc1 numc2;
array fixmc1 (t) fixmc11 fixmc12;
array fixmc2 (t) fixmc21 fixmc22;
array fixmc3 (t) fixmc31 fixmc32;
array fixmm1 (t) fixmm11 fixmm12;
array fixmm2 (t) fixmm21 fixmm22;
array fixmm3 (t) fixmm31 fixmm32;
array fixyc1 (t) fixyc11 fixyc12;
array fixyc2 (t) fixyc21 fixyc22;
array fixyc3 (t) fixyc31 fixyc32;
array fixym1 (t) fixym11 fixym12;
array fixym2 (t) fixym21 fixym22;
array fixym3 (t) fixym31 fixym32;
array ysc (p) ysca yscb ysc;
array msc (p) msca mscb msc;
array yec (p) yeca yecb yecc;
array mec (p) meca mecb mecc;
array ymsc (p) ymsca ymscb ymscc;
array ymec (p) ymeca ymecb ymec;
array csmc (p) csmca csmcb csmcc;
array cemc (p) cemca cemcb cemcc;
array cdli (p) cdli1 cdlib cdlic;
array cbeg (p) cbega cbegb cbegc;
array cbeg2 (p) cbeg2a cbeg2b cbeg2c;
array mbeg (p) mbega mbegb mbegc;
array cmars (p) cmarsa cmarsb cmarsc;
array ymsm (p) ymsma ymsmb ymsmc;
array csmm (p) csmma csmb csmmc;
array csms (p) csmsa csmsb csmsc;
array ymss (p) ymssa ymssb ymssc;
array cemm (p) cemma cemmb cemmc;
array cems (p) cemsa cemsb cemsc;
array how1 (p) howa1 howb1 howc1;
array how2 (p) howa2 howb2 howc2;
array how3 (p) howa3 howb3 howc3;
array how4 (p) howa4 howb4 howc4;
array how5 (p) howa5 howb5 howc5;
array hym1 (p) hyma1 hymb1 hymc1;
array hmm1 (p) hmma1 hmmb1 hmmc1;
array hym2 (p) hyma2 hymb2 hymc2;
array hmm2 (p) hmma2 hmmb2 hmmc2;

array hym3 (p) hyma3 hymb3 hymc3;
array hmm3 (p) hmma3 hmmb3 hmmc3;
array hym4 (p) hyma4 hymb4 hymc4;
array hmm4 (p) hmma4 hmmb4 hmmc4;
array hys1 (p) hysa1 hysb1 hysc1;
array hms1 (p) hmsa1 hmsb1 hmsc1;
array hys2 (p) hysa2 hysb2 hysc2;
array hms2 (p) hmsa2 hmsb2 hmsc2;
array hys3 (p) hysa3 hysb3 hysc3;
array hms3 (p) hmsa3 hmsb3 hmsc3;
array hys4 (p) hysa4 hysb4 hysc4;
array hms4 (p) hmsa4 hmsb4 hmsc4;
array mdli (p) mdlia mdlib mdlic;
array nuc (p) nuca nucb nucc;
array num (p) numa numb numc;
array fixmc (p) fixmc1 fixmc2 fixmc3;
array fixmm (p) fixmm1 fixmm2 fixmm3;
array fixyc (p) fixyc1 fixyc2 fixyc3;
array fixym (p) fixym1 fixym2 fixym3;

if M54011_1>-4 then M45001_1=M54011_1;
if M54012_1>-4 then M45002_1=M54012_1;

do p=1 to 3;
do t=1 to 2;
  if cdli=1 then ymsc=dliy;
  if mdli=1 then do; cmars=1; ymsm=dliy; end;

/*hand edit*/
  if mdli=3 and pubid ne 4303 then do;
    cmars=2; ymss=dliy; end;
    if -2 le mdli le -1 then cmars=mdli;
    if ysc>0 and msc>0 then ymsc=(ysc*100)+msc;
    if -3 le ysc le 0 then do; fixyc=1; end;
    if -3 le msc le 0 then do; fixmc=1; end;
    if ysc>0 and msc>0 then ymsc=(ysc*100)+msc;
    if -3 le ysc le 0 or -3 le msc le 0 then do;
      ymsc=dliy; end;

/*need to add stop information for arrays*/
  if yec>0 and mec>0 then ymec=(yec*100)+mec;
  if -3 le yec le 0 or -3 le mec le 0 and cbeg ne 1 and
    cdli ne 1 and cbeg2 ne 1 then do;
    ymec=iym; end;
  if -3 le yec le 0 or -3 le mec le 0 and (cbeg=1 or
    cdli=1 or cbeg2=1) then do;
    ymec=iym-1; end;

  csmc=(round(ymsc,100)-198000)*.12+(ymsc-
    round(ymsc,100));
  cemc=(round(ymec,100)-198000)*.12+(ymec-
    round(ymec,100));
  if cbeg=1 or cdli=1 or cbeg2=1 then cemc=doicm;

  if mbeg=1 then do; cmars=1; ymsm=ymsc; ymsc=.;
    if fixmc=1 then fixmm=1;
    if fixyc=1 then fixym=1;

```

```

end;

if how1=1 and hym1>0 and hmm1>0 then do;
  cmars=1; ymsm=(hym1*100)+hmm1; end;
if -2 le how1 le -1 then cmars=how1;

if how2=1 and hym2>0 and hmm2>0 then do;
  cmars=1; ymsm=(hym2*100)+hmm2; end;
if how2=3 and hys2>0 and hms2>0 then do;
  cmars=2; ymss=(hys2*100)+hms2; end;
if how2=4 and hys2>0 and hms2>0 then do;
  cmars=3; ymss=(hys2*100)+hms2; end;
if how2=5 and hys2>0 and hms2>0 then do;
  cmars=0; ymss=(hys2*100)+hms2; end;
if -2 le how2 le -1 then cmars=how2;

if how3=1 and hym3>0 and hmm3>0 then do;
  cmars=1; ymsm=(hym3*100)+hmm3; end;
if how3=3 and hys3>0 and hms3>0 then do;
  cmars=2; ymss=(hys3*100)+hms3; end;
if -2 le how3 le -1 then cmars=how3;

if how4=1 and hym4>0 and hmm4>0 then do;
  cmars=1; ymsm=(hym4*100)+hmm4; end;
if how4=3 and hys4>0 and hms4>0 then do;
  cmars=2; ymss=(hys4*100)+hms4; end;
if -2 le how4 le -1 then cmars=how4;

if how5=1 and hym5>0 and hmm5>0 then do;
  cmars=1; ymsm=(hym5*100)+hmm5; end;
if how5=3 and hys5>0 and hms5>0 then do;
  cmars=2; ymss=(hys5*100)+hms5; end;
if -2 le how5 le -1 then cmars=how5;

csmm=(round(ymsm,100)-198000)*.12+(ymsm-
  round(ymsm,100));
csms=(round(ymss,100)-198000)*.12+(ymss-
  round(ymss,100));
if csms>0 then cems=doicm;
if csmm>0 then cemm=doicm;

if csmc>0 then do;
  tlc=tlc+1; nuc=tlc;
  if nuc=1 then cohcm=csmc;
end;

if csmm>0 then do;
  ttlm=tlm+1; num=ttl;
  if num=1 then marcm=csmm;
end;

corrc=0; corrm=0;
if tlc>-1 or ttlm>-1 then do;
  if csmca2>0 and csmca2 ne . then do;
    corrc=corrc+1; end;
  if csmcb2>0 and csmcb2 ne . then do;
    corrc=corrc+1; end;

```

```

if csmcc2>0 and csmcc2 ne . then do;
  corrc=corrc+1; end;
if csmma2>0 and csmma2 ne . then do;
  corrm=corrm+1; end;
if csmb2>0 and csmb2 ne . then do;
  corrm=corrm+1; end;
if csmmc2>0 and csmmc2 ne . then do;
  corrm=corrm+1; end;
end;

ttlcnew=ttlc-corrc; ttlmnew=tlm-corrm;

C=0;
do L=1 to 232;
  C=C+1;
  if csmc>0 and cemc>0 and csmc LE C LE cemc
    then coh=1;
  if cmars=1 and csmm>0 and cemm>0 and csmm
    le c le cemm then m=1;
  if 2 le cmars le 3 and csms>0 and cems>0 and
    csms le c le cems then m=cmars;
  if -2 le cmars le -1 then m=cmars;
  if c=doicm then do;
    if -2 le m le -1 then mars=m;
    if m=0 and coh=1 then mars=1;
    if m=0 and coh=0 then mars=2;
    if m=1 and coh=1 then mars=3;
    if m=1 and coh=0 then mars=4;
    if m=2 and coh=1 then mars=5;
    if m=2 and coh=0 then mars=6;
    if m=3 and coh=1 then mars=7;
    if m=3 and coh=0 then mars=8;
    if m=4 and coh=1 then mars=9;
    if m=4 and coh=0 then mars=10;
    marstat=mars;
  end;
end;
end;
end;
end;

if 1 le marstat le 2 then cmarstat=0;
if 3 le marstat le 4 then cmarstat=1;
if 5 le marstat le 6 then cmarstat=2;
if 7 le marstat le 8 then cmarstat=3;
if 9 le marstat le 10 then cmarstat=4;
if -2 le marstat le -1 then cmarstat=marstat;

array cvcm cohcm marcm;
array cvy cohy mary;
array cvm cohm marm;

do over cvcm;
  if 229 le cvcm le 232 then do;
    cvy=1999; cvm=cvcm-228; end;
  if 217 le cvcm le 228 then do;
    cvy=1998; cvm=cvcm-216; end;

```

Appendix 3: Family Background and Formation Variable Creation

```

if 205 le cvcm le 216 then do;
  cvy=1997; cvm=cvcm-204; end;
if 193 le cvcm le 204 then do;
  cvy=1996; cvm=cvcm-192; end;
if 181 le cvcm le 192 then do;
  cvy=1995; cvm=cvcm-180; end;
if 169 le cvcm le 180 then do;
  cvy=1994; cvm=cvcm-168; end;
if cvcm=. then do; cvcm=-4; cvy=-4; cvm=-4; end;
end;

/*hand edits for those with start date prior to 1994*/
if pubid=3927 then do; cohlm=1; cohyl=1981; end;
if pubid=5848 then do; cohlm=12; cohyl=1980; end;
if pubid=7891 then do;
  cohlm=12; cohyl=1981; marm=12; mary=1981; end;
if pubid=8522 then do; cohlm=9; cohyl=1981; end;

/*to correct for those who don't know the start month
for cohabiting*/
if fixmc11=1 and (m31001m<0 and m31001m>-4)
then do;
  cohym=m31001y; cohcm=-3; cohlm=-3; end;
/*to correct for those who don't know the start year for
cohabiting*/
if fixyc11=1 and (m31001y<0 and m31001y>-4) then
do;
  cohym=m31001m; cohcm=-3; cohyl=-3; end;
/*to correct for those who don't know either the start
year or month for cohabiting*/
if fixyc11=1 and fixmc11=1 then do;
  cohym=-3; cohcm=-3; cohyl=-3; end;

/*correct for those who don't know start month for
marriage*/
if fixmm11=1 and (m31001y<0 and m31001y>-4)
then do;
  mary=m31001y; marcm=-3; marm=-3; end;
/*correct for those who don't know start year for
marriage*/
if fixym11=1 and (m31001m<0 and m31001m>-4)
then do;
  marm=m31001m; marcm=-3; mary=-3; end;
/*to correct for those who don't know either the start
year or month for marriage*/
if fixym11=1 and fixmm11=1 then do;
  marm=-3; marcm=-3; mary=-3; end;

/*if old date present and don't deny then use old date*/
if fcohym ne -4 and (m712=1 or m714=1 or (m712=0
and m718=1)) then do;
  cohcm=fcohym; cohyl=fcoh_y; cohlm=fcoh_m; end;
if fmarm ne -4 and (m712=1 or m714=1) then do;
  marcm=fmarm; mary=fmar_y; marm=fmar_m; end;

/*correct total marriages & cohabs to reflect previous*/
/*previously in relationship, none since interview*/
if m712=1 and m740=0 then do;
  ttlc=coh_ttl;
  if mar_ttl ne 0 then ttlmnew=mar_ttl;
end;
/*previously in relationship, in more since interview*/
if m712=1 and m740=1 then do;
  ttlmnew=ttlmnew+mar_ttl;
  ttlcnew=ttlcnew+coh_ttl;
end;
/*in relationship at int date, no more since interview*/
if (m714=1 or m716=1) and m730=0 then do;
  if mar_ttl ne 0 then ttlmnew=mar_ttl;
  ttlcnew=coh_ttl;
end;
/*in relationship at int date, in more since interview*/
if (m714=1 or m716=1) and m730=1 then do;
  ttlmnew=ttlmnew+mar_ttl;
  ttlcnew=ttlcnew+coh_ttl;
end;
/*hand edits for separation without marriage*/
if pubid=3717 or pubid=5848 then do;
  marstat=1; cmarstat=0;
end;

if m700=-4 then do;
  marstat=-4; cmarstat=-4; ttlmnew=-4;
  ttlcnew=-4; cohcm=-4; marcm=-4;
  cohyl=-4; mary=-4; cohlm=-4; marm=-4;
end;

if m700=-5 then do;
  marstat=-5; cmarstat=-5; ttlmnew=-5;
  ttlcnew=-5; cohcm=-5; marcm=-5;
  cohyl=-5; mary=-5; cohlm=-5; marm=-5;
end;

/*hand edits based on interviewer remarks*/
if pubid=39 or pubid=6025 then do;
  marstat=2; cmarstat=0; ttlmnew=0;
  ttlcnew=0; cohcm=-4; marcm=-4;
  cohyl=-4; mary=-4; cohlm=-4; marm=-4;
end;

value fcmar 0 ='never married' 1 ='married'
2 ='separated' 3 ='divorced' 4 ='widowed';
value fcmmp 169-192='94-5' 193-216='96-7' 217-
232='98-9';

array m norcid cohym cohyl cohcm marm mary marcm
marstat cmarstat ttlcnew ttlmnew;
array X X01-X11;

do over X; X=m; if X=. then X=-4; end;
endsas;

```

YOUTH'S FERTILITY AND CHILD STATUS

Variables Created:	CV_CHILD_BIRTH_DATE.xx_M CV_CHILD_DEATH_DATE.xx_M CV_CHILD_BIRTH_MONTH.xx CV_CHILD_STATUS.xx CV_BIO_CHILD_HH	CV_CHILD_BIRTH_DATE.xx_Y CV_CHILD_DEATH_DATE.xx_Y CV_CHILD_DEATH_MONTH.xx CV_BIO_CHILD_NR
---------------------------	--	--

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
BDEAD971, BDEAD972	BIOCHILD_DEAD.01, .02	F59001, F59002	YFER-5900.01, .02
CVBIRM1, CVBIRY1	CV_CHILD_BIRTH_DATE.01_M, _Y	F6000M, F6000Y	YFER-6000.01~M, ~Y
CVBIRM2, CVBIRY2	CV_CHILD_BIRTH_DATE.02_M, _Y	BDAYM1, BDAYY1	BIOCHILD_BDATE.01-M, ~Y
CVDEAM1, CVDEAY1	CV_CHILD_DEATH_DATE.01_M, _Y	BDAYM2, BDAYY2	BIOCHILD_BDATE.02-M, ~Y
CVSTAT1, CVSTAT2	CV_CHILD_STATUS.01, .02	BDAYM3, BDAYY3	BIOCHILD_BDATE.03-M, ~Y
F400	YFER-400	BDEAD1-BDEAD3	BIOCHILD_DEAD.01-.03
F56001M, F56001Y	YFER-5600.01~M, ~Y	RES1-RES3	BIOCHILD_RESIDE.01-.03
F56002M, F56002Y	YFER-5600.02~M, ~Y	PUBID	PUBID

Codes for Created Variables

Date of birth and death variables
Date variables are presented as both the actual month and year and the month number in a continuous month scheme.

Status variables
1 Adopted
2 Deceased
3 Non-resident, foster care
4 Non-resident, not adopted or in foster care
5 Resident

This program creates a number of variables describing the youth's fertility and the current status of the youth's children. For more information on the continuous month system, see appendix 7 in this document.

```
/* first, we create a variable indicating the dobm(i) and
doby(i) for each biological child. Information is taken
from the fertility roster(BIOC) and the fertility section
of the youth survey (YFER). */

array bdaym[3] bdaym1-bdaym3;
array bdayy[3] bdayy1-bdayy3;
array dobm[3] dobm1-dobm3 ;
array doby[3] doby1-doby3 ;

do i=1 to 3;
  dobm(i)=-4;
  if bdaym(i) eq -3 then dobm(i)=-3;
  if bdaym(i) gt 0 then dobm(i)=bdaym(i);
end;

do i=1 to 3;
  doby(i)=-4;
  if bdayy(i) eq -3 then doby(i)=-3;
  if bdayy(i) gt 0 then doby(i)=bdayy(i);
end;

/* second, create a continuous month scheme variable
for the month of birth of the children using the
formula: (12*(doby(i)-1980)+dobm(i)) */
```

```
array mob[3] mob1-mob3 ;

do i=1 to 3;
  mob(i)=-4;
  if dobm(i) eq -3 or doby(i) eq -3 then mob(i)=-3;
  if dobm(i) gt 0 and doby(i) ge 1980 then
    mob(i)=12*(doby(i)-1980)+dobm(i);
end;

/* third, create an actual date variable for the date of
death of the youth's children */

array bdead[3] bdead1-bdead3;
array dodm[3] dodm1-dodm3;
array dody[3] dody1-dody3;

do i=1 to 3;
  dodm(i)=-4;
  dody(i)=-4;
end;

if cvdeam1 gt 0 then dodm1=cvdeam1;
if cvdeam1=-3 then dodm1=-3;
if f6000m gt 0 then dodm1=f6000m;

if cvdeay1 gt 0 then dody1=cvdeay1;
```

Appendix 3: Family Background and Formation Variable Creation

```

if cvdeay1=-3 then dody1=-3;
if f6000y gt 0 then dody1=f6000y;

/* fourth, create a continuous month scheme variable
for the month of death of the children using the
formula: (12*(dody(i)-1980)+dodm(i)) */

array mod[3] mod1-mod3;

do i=1 to 3;
  if dodm(i)=-4 then mod(i)=-4;
  if dodm(i)=-3 then mod(i)=-3;
  if dodm(i) gt 0 and dody(i) ge 1980 then
    mod(i)=12*(dody(i)-1980)+dodm(i);
end;

/* taking care of two special cases, where the listed
children were determined not to exist.*/
if pubid=1714 or pubid=6734 then do;
  do i=1 to 3;
    dobmi=-4; doby(i)=-4; mob(i)=-4;
    dodm(i)=-4; dody(i)=-4; mod(i)=-4;
  end;
end;

/* fifth, create a variable indicating the status of youth's
first (second, third) child: */

array res[3] res1-res3;
array status[3] status1-status3 ;

/*initialize the status variable and determine which kids
live at home */
do i=1 to 3;
  status(i)=-4;
  if mob(i) eq -4 or res(i) eq -4 or res(i)=-5 then
    status(i)=-4;
  if res(i) eq -3 then status(i)=-3;
  if res(i) eq 1 then status(i)=5;
end;

/* if they are dead */
do i=1 to 3;
  if dodm(i) gt 0 then status(i)=2;
end;

/* res(i)=0: children's status was not updated in some
cases. F5900 variables are used to determine the status
for some of those children. */

/* adjusted by f59001 variable*/
do i=1 to 3;
  if res(i)=0 and dodm(i)=-4 then do;
    status(i)=-3;
  end;

if bdayy(i) = f56001y and bdaym(i) = f56001m then
  do;
    if f59001 gt 0 then status(i)=4;
    if f59001=1 then status(i)=5;
    if f59001=3 then status(i)=1;
    if f59001=4 then status(i)=3;
  end;
end;

/* adjusted by f59002 variable*/
do i=1 to 3;
  if bdayy(i) = f56002y and bdaym(i) = f56002m then
    do;
      if f59002 gt 0 then status(i)=4;
      if f59002=1 then status(i)=5;
      if f59002=3 then status(i)=1;
      if f59002=4 then status(i)=3;
    end;
end;

/* sixth, the number of children ever born and residing
in the household (tbiores) */

array biores[3] bires1-bires3;

/*initialize the bires variable and create tbiores */
do i=1 to 3;
  bires(i)=0;
  if status(i) eq 5 then bires(i)=1;
end;

tbiores=bires1+bires2+bires3;
if mob1=-4 then tbiores=-4;

/* seventh, the number of children ever born and not
residing in the household (tbionres) */

array bionres[3] bionres1-bionres3;

do i=1 to 3;
  bionres(i)=0;
  if (status(i) eq 1 or status(i) eq 3 or status(i) eq 4 or
      res(i)=0) and status(i) ne 2 then bionres(i)=1;
end;

tbionres=bionres1+bionres2+bionres3;
if mob1=-4 then tbionres=-4;

/* sort created variables by birthdays, so that the first
child listed is the oldest child. */
rmob1=0;
rmob2=0;
rmob3=0;

array rmob[3] rmob1 rmob2 rmob3;
array m[3] m1 m2 m3;

```

```

do i=1 to 3;
    rmob(i)=mob(i);
end;

do i=1 to 3;
    if rmob(i)=-3 then mob(i)=1000;
end;

m1=-4; m2=-4; m3=-4;

/* consider the families with one child */
if (mob1>-4 and mob2=-4 and mob3=-4) or (mob1=-4
    and mob2>-4 and mob3=-4) or (mob1=-4 and
    mob2=-4 and mob3>-4) then do;
    if mob1>-4 then m1=mob1;
    if mob2>-4 then m1=mob2;
    if mob3>-4 then m1=mob3;
end;

/* consider families with 2 children */
if (mob1>-4 and mob2>-4 and mob3=-4) then do;
    if mob1 ge mob2 then do;
        m1=mob2; m2=mob1;
    end;
    if mob1<mob2 then do;
        m1=mob1; m2=mob2;
    end;
end;

if (mob1=-4 and mob2>-4 and mob3>-4) then do;
    if mob2 ge mob3 then do;
        m1=mob3; m2=mob2;
    end;
    if mob2<mob3 then do;
        m1=mob2; m2=mob3;
    end;
end;

if (mob1>-4 and mob2=-4 and mob3>-4) then do;
    if mob1 ge mob3 then do;
        m1=mob3; m2=mob1;
    end;
    if mob1<mob3 then do;
        m1=mob1; m2=mob3;
    end;
end;

/* consider families with three children */
if (mob1>-4 and mob2>-4 and mob3>-4) then do;
    m1=min(mob1, mob2, mob3);
    m3=max(mob1, mob2, mob3);
end;
do i=1 to 3;
    if mob(i) ne m1 and mob(i) ne m3 then m2=mob(i);
end;

array cvbm[3] cvbm1-cvbm3;
array cvby[3] cvby1-cvby3;
array cvmob[3] cvmob1-cvmob3;
array cvdm[3] cvdm1-cvdm3;
array cvdy[3] cvdy1-cvdy3;
array cvmod[3] cvmod1-cvmod3;
array cvstat[3] cvstat1-cvstat3;

do i=1 to 3;
    if m1=mob(i) then do;
        cvbm1=dobm(i); cvby1=doby(i); cvmob1=rmob(i);
        cvdm1=dodm(i); cvdy1=dody(i); cvmod1=mod(i);
        cvstat1=status(i);
    end;
end;

do i=1 to 3;
    if m2=mob(i) then do;
        cvbm2=dobm(i); cvby2=doby(i); cvmob2=rmob(i);
        cvdm2=dodm(i); cvdy2=dody(i); cvmod2=mod(i);
        cvstat2=status(i);
    end;
end;

do i=1 to 3;
    if m3=mob(i) then do;
        cvbm3=dobm(i); cvby3=doby(i); cvmob3=rmob(i);
        cvdm3=dodm(i); cvdy3=dody(i); cvmod3=mod(i);
        cvstat3=status(i);
    end;
end;

/* non-interview cases*/
if f400=-5 then do;
    do i=1 to 3;
        cvbm[i]=-5; cvby[i]=-5; cvmob[i]=-5;
        cvdm[i]=-5; cvdy[i]=-5; cvmod[i]=-5;
        cvstat[i]=-5;
    end;
    tbiores=-5; tbionres=-5;
end;

/* special case—misidentification for one baby */
if pubid=8729 then do;
    cvbm1=11; cvby1=1992; cvmob1=155;
    cvdm1=11; cvdy1=1996; cvmod1=203;
    cvstat1=2;
    cvbm2=2; cvby2=1996; cvmob2=194;
    cvdm2=-4; cvdy2=-4; cvmod2=-4;
    cvstat2=5;
    cvbm3=9; cvby3=1998; cvmob3=225;
    cvdm3=-4; cvdy3=-4; cvmod3=-4;
    cvstat3=5;
    tbiores=2; tbionres=0;
end;

endsas;

```

NUMBER OF RESIDENCES SINCE AGE 12

Variables Created: CV_TTL_RESIDENCES

Variables Used

Name in Program	Question Name on CD
CV	CV_TTL_RESIDENCES
Y3500	YHHI-3500
Y3600	YHHI-3600

This program calculates the number of residences in which youth has lived since age 12. In round 1, the variable was created with information from the parent interview. In round 2, information collected from the respondent is combined with the round 1 variable to update the previous information.

```
/* Initialize each case to -4 */

RESID=-4;

/* Refusals in either round1 or round2 */
if cv=-1 or y3600=-1 then resid=-1;

/* The answer is don't know in either round1 or round2 */
if cv=-2 or y3600=-2 then resid=-2;

/* Non-negative answers in both round, then add up the numbers of residences in two rounds */
if cv ge 0 and y3600 ge 0 then resid=cv+y3600;
if cv ge 0 and y3500=0 then resid=cv;

/* If parents were not interviewed in round1*/
if cv=-4 then resid=-3;

/* If respondents were not interviewed in round2 */
if Y3600=-5 then resid=-5;

endsas;
```